M2 MPRI - Programmation Probabiliste - Discrete semantics

Christine Tasson

Exercice 1. *Equations*

Démontrer que les programmes suivants ont la même sémantique en vous aidant de la Figure 1

```
def coin():                          ≡   def coin():
    x = sample Bernoulli(0.5)                x = sample(Binomial(2n, 0.5))
    return x                                 return x > n
```

```
assume(a); ... ; assume(b)           ≡   ... ; assume (a and b)
```

```
x = sample(d); y = sample(d')        ≡   y = sample(d'); x = sample(d)
```

Exercice 2. *Coin*

Calculer la sémantique des programmes suivants.

```
def flip(p):
    x = sample(Bernoulli(p), name='x')
    y = sample(Bernoulli(p), name='y')
    assume(x!=y)
    return x

def stop(p:float):
    n = 0
    while sample(Bernoulli(p)):
        n = n+1
    return n
```

Exercice 3.

Traduisez en pPCF, puis calculer la sémantique opérationnelle et la sémantique dénotationnelle des programmes suivants écrits dans le style mu-PPL.

```
def coin_reject(d: Dist[bool]):
    x = sample(d
    y = sample(d)
    return x if (x!=y) else coin_reject(d)

def geo(d:Dist[bool]):
    x = sample(d, nam)
    return geo(d, s+'_') if x else True
```

Syntaxe :

$$t ::= \texttt{None} \mid \texttt{bool} \mid \texttt{int} \mid t \times t \mid \texttt{dist}(t)$$

$$e ::= c \mid x \mid \texttt{(}e\texttt{, }e\texttt{)} \mid op\texttt{(}e\texttt{)}$$

$$s ::= \texttt{pass} \mid x \texttt{ = } e \mid x \texttt{ = } f\texttt{(}e\texttt{)} \mid \texttt{if } e\texttt{: } s \texttt{ else: } s \mid s\texttt{; } s \mid x \texttt{ = sample(}e\texttt{)} \mid \texttt{assume(}e\texttt{)}$$

$$d ::= \texttt{def } f\texttt{(}x\texttt{): } s \texttt{ return } e \mid d\ d$$

Types :

$[\![t]\!]$	$=$	t	*Countable set*
$[\![\texttt{None}]\!]$	$=$	$\{*\}$	*one point set*
$[\![\texttt{bool}]\!]$	$=$	$\mathbb{B}$	*two points* ***True*** *et* ***False*** *sets*
$[\![\texttt{int}]\!]$	$=$	$\mathbb{N}$	*integers*
$[\![t_1 \times t_2]\!]$	$=$	$t_1 \times t_2$	*products*
$[\![\texttt{dist}(t)]\!]$	$=$	$t \to \mathbb{R}^+$	*value to score tables*

Expressions :

$$\begin{array}{lcl} [\![c]\!]^{\varphi}_{\gamma} & = & c \\ [\![x]\!]^{\varphi}_{\gamma} & = & \gamma(x) \\ [\![op\texttt{(}e\texttt{)}]\!]^{\varphi}_{\gamma} & = & op([\![e]\!]^{\varphi}_{\gamma}) \\ [\![\texttt{(}e_1\texttt{, }e_2\texttt{)}]\!]^{\varphi}_{\gamma} & = & ([\![e_1]\!]^{\varphi}_{\gamma}, [\![e_2]\!]^{\varphi}_{\gamma}) \end{array}$$

Commandes :

$$[\![\texttt{pass}]\!]^{\varphi}_{\gamma}(\gamma') = \delta_{\gamma}(\gamma') = 1 \text{ if } \gamma = \gamma" \text{ and } 0 \text{ otherwise}$$

$$[\![x \texttt{ = } e]\!]^{\varphi}_{\gamma}(\gamma') = \delta_{\gamma + \left[x \leftarrow [\![e]\!]^{\varphi}_{\gamma}\right]}(\gamma')$$

$$[\![\texttt{if } e\texttt{: } s_1 \texttt{ else: } s_2]\!]^{\varphi}_{\gamma} i(\gamma') = [\![s_1]\!]^{\varphi}_{\gamma}(\gamma') \ if [\![e]\!]^{\varphi}_{\gamma} \ otherwise [\![s_2]\!]^{\varphi}_{\gamma}(\gamma')$$

$$[\![x \texttt{ = sample(}e\texttt{)}]\!]^{\varphi}_{\gamma}(\gamma') = \sum_{v} [\![e]\!]^{\varphi}_{\gamma}(v)\ \delta_{\gamma + [x \leftarrow v]}(\gamma')$$

$$[\![x \texttt{ = } f\texttt{(}e\texttt{)}]\!]^{\varphi}_{\gamma}(\gamma') = \sum_{v} \mu(v)\ \delta_{\gamma + [x \leftarrow v]}(\gamma') with\ \mu = \varphi(f)([\![e]\!]^{\varphi}_{\gamma})$$

$$[\![\texttt{assume(}e\texttt{)}]\!]^{\varphi}_{\gamma}(\gamma') = \delta_{\texttt{True}}([\![e]\!]^{\varphi}_{\gamma})\ \delta_{\gamma}(\gamma')$$

$$[\![s_1\texttt{; } s_2]\!]^{\varphi}_{\gamma}(\gamma') = \sum_{\gamma_1} [\![s_1]\!]^{\varphi}_{\gamma}(\gamma_1)\ [\![s_2]\!]^{\varphi}_{\gamma_1}(\gamma')$$

Déclarations et inférence :

$$[\![\texttt{def } f\texttt{(}x\texttt{): } s \texttt{ return } e]\!]^{\varphi} = \varphi + [f \leftarrow \lambda v.\ \lambda v'.\ \theta(v, v')] \quad with\ \theta(v, v') = \sum_{\gamma} [\![s]\!]^{\varphi}_{[x \leftarrow v]}(\gamma)\ \delta_{[\![e]\!]^{\varphi}_{\gamma}}(v')$$

$$[\![d_1\ d_2]\!]^{\varphi} = [\![d_2]\!]^{\varphi_1} \ with\ \varphi_1 = [\![d_1]\!]^{\varphi}$$

$$[\![\texttt{infer(}f\texttt{, }e\texttt{)}]\!]^{\varphi}_{\gamma}(v) = \begin{cases} \dfrac{\mu(v)}{\mu(\top)} & if\ 0 < \mu(\top) < \infty \\ \text{Error} & otherwise \end{cases}$$
$$with\ \mu(v) = \varphi(f)([\![e]\!]^{\varphi}_{\gamma})$$
$$and\ \mu(\top) = \textstyle\sum_v \mu(\top)$$

Figure 1 : Sémantique de pIMP.